Cray XT Series System Overview

S–2423–15

© 2004–2006 Cray Inc. All Rights Reserved. This manual or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Cray Inc.

U.S. GOVERNMENT RESTRICTED RIGHTS NOTICE

The Computer Software is delivered as "Commercial Computer Software" as defined in DFARS 48 CFR 252.227-7014.

All Computer Software and Computer Software Documentation acquired by or for the U.S. Government is provided with Restricted Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7014, as applicable.

Technical Data acquired by or for the U.S. Government, if any, is provided with Limited Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7013, as applicable.

Cray, LibSci, UNICOS and UNICOS/mk are federally registered trademarks and Active Manager, Cray Apprentice2, Cray C++ Compiling System, Cray Fortran Compiler, Cray SeaStar, Cray SHMEM, Cray X1, Cray X1E, Cray XD1, Cray XT3, CrayDoc, CRInform, Libsci, RapidArray, UNICOS/lc, and UNICOS/mp are trademarks of Cray Inc.

AMD is a trademark of Advanced Micro Devices, Inc. Copyrighted works of Sandia National Laboratories include: Catamount/QK, Compute Processor Allocator (CPA), and xtshowmesh. GCC is a trademark of the Free Software Foundation, Inc.. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. Linux is a trademark of Linus Torvalds. Lustre was developed and is maintained by Cluster File Systems, Inc. under the GNU General Public License. MySQL is a trademark of MySQL AB. Opteron is a trademark of Advanced Micro Devices, Inc. PBS Pro is a trademark of Altair Grid Technologies. SUSE is a trademark of SUSE LINUX Products GmbH, a Novell business. The Portland Group and PGI are trademarks of STMicroelectronics. TotalView is a trademark of Etnus, LLC.

Record of Revision

Version	*Description*
1.0	December 2004 Draft documentation to support Cray XT3 early-production systems.
1.0	March 2005 Draft documentation to support Cray XT3 limited-availability systems.
1.1	June 2005 Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.1, System Management Workstation (SMW) 1.1, and UNICOS/lc 1.1 releases.
1.2	August 2005 Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.2, System Management Workstation (SMW) 1.2, and UNICOS/lc 1.2 releases.
1.3	November 2005 Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.3, System Management Workstation (SMW) 1.3, and UNICOS/lc 1.3 releases.
1.4	April 2006 Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.4, System Management Workstation (SMW) 1.4, and UNICOS/lc 1.4 releases.
1.5	August 2006 Supports limited availability (LA) release of Cray XT series systems running the Cray XT series Programming Environment 1.5, UNICOS/lc 1.5, and System Management Workstation 1.5 releases.
1.5	October 2006 Supports general availability (GA) release of Cray XT series systems running the Cray XT series Programming Environment 1.5, UNICOS/lc 1.5, and System Management Workstation 1.5 releases.

Contents

Page

Preface **vii**

Accessing Product Documentation vii

Conventions viii

Reader Comments ix

Cray User Group ix

Introduction [1] **1**

Related Publications 5

Publications for Application Developers 5

Publications for System Administrators 6

Hardware Overview [2] **9**

Basic Hardware Components 9

AMD Opteron Processor 9

DIMM Memory 12

Cray SeaStar Chip 13

System Interconnection Network 14

Nodes 14

Compute Nodes 15

Service Nodes 15

Blades, Chassis, and Cabinets 16

Blades 16

Chassis and Cabinets 17

I/O System 18

Software Overview [3] **21**

UNICOS/lc Operating System 21

Page

Lustre File System 22
Cray XT Series Development Environment 22
User Environment 23
System Access 23
Compilers 23
PGI Compiler Commands 24
GCC Compiler Commands 24
Libraries 25
Linker 26
Runtime Environment 27
Interactive Jobs 27
Batch Jobs 28
Debugging Applications 30
Measuring Performance 31
Performance API (PAPI) 31
CrayPat 31
Cray Apprentice2 32
System Administration 33
System Management Workstation (SMW) 33
Shared-Root File System 33
Lustre File System Administration 34
Lustre UFS-Like File System 35
Lustre Parallel File System 36
Configuration and Source Files 37
System Log 37
Service Database (SDB) 37
System Activity Reports 38

Cray RAS and Management System (CRMS) [4] 39

CRMS Hardware 39
CRMS Network 40

Page

System Management Workstation 41
Hardware Controllers 41
CRMS Software 41
Software Monitors 41
CRMS Administrator Interfaces 42
CRMS Actions 43
System Startup and Shutdown 43
Event Probing 43
Event Logging 44
Event Handling 44

Glossary **45**

Index **51**

Figures

Figure 1. Cray XT Series Supercomputer System 4
Figure 2. Single-core Processor 10
Figure 3. Dual-core Processor 12
Figure 4. Cray SeaStar Chip 14
Figure 5. Cray XT Series System Compute Node 15
Figure 6. Cray XT Series System Service Node 16
Figure 7. Chassis and Cabinet (front view) 17
Figure 8. I/O System 19
Figure 9. Launching Interactive Jobs 28
Figure 10. Launching Batch Jobs 30
Figure 11. Lustre UFS-Like File System 35
Figure 12. Lustre Parallel File System 36
Figure 13. CRMS Components 40

Preface

The information in this preface is common to Cray documentation provided with this software release.

Accessing Product Documentation

With each software release, Cray provides books and man pages, and in some cases, third-party documentation. These documents are provided in the following ways:

CrayDoc

The Cray documentation delivery system that allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. Access this HTML and PDF documentation via CrayDoc at the following locations:

- The local network location defined by your system administrator
- The CrayDoc public website: `docs.cray.com`

Man pages

Access man pages by entering the `man` command followed by the name of the man page. For more information about man pages, see the `man`(1) man page by entering:

```
% man man
```

Third-party documentation

Access third-party documentation not provided through CrayDoc according to the information provided with the product.

Conventions

These conventions are used throughout Cray documentation:

Convention	Meaning
`command`	This fixed-space font denotes literal items, such as file names, pathnames, man page names, command names, and programming language elements.
variable	Italic typeface indicates an element that you will replace with a specific value. For instance, you may replace *filename* with the name `datafile` in your program. It also denotes a word or concept being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a syntax representation for a command, library routine, system call, and so on.
. . .	Ellipses indicate that a preceding element can be repeated.
`name`(N)	Denotes man pages that provide system and programming reference information. Each man page is referred to by its name followed by a section number in parentheses. Enter: `% man man` to see the meaning of each section number for your particular system.

Reader Comments

Contact us with any comments that will help us to improve the accuracy and usability of this document. Be sure to include the title and number of the document with your comments. We value your comments and will respond to them promptly. Contact us in any of the following ways:

E-mail:
`docs@cray.com`

Telephone (inside U.S., Canada):
1–800–950–2729 (Cray Customer Support Center)

Telephone (outside U.S., Canada):
+1–715–726–4993 (Cray Customer Support Center)

Mail:
Customer Documentation
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Cray User Group

The Cray User Group (CUG) is an independent, volunteer-organized international corporation of member organizations that own or use Cray Inc. computer systems. CUG facilitates information exchange among users of Cray systems through technical papers, platform-specific e-mail lists, workshops, and conferences. CUG memberships are by site and include a significant percentage of Cray computer installations worldwide. For more information, contact your Cray site analyst or visit the CUG website at `www.cug.org`.

Introduction [1]

This document provides an overview of Cray XT series systems. The intended audience is the application developer and system administrator. Prerequisite knowledge is a familiarity with the concepts of high-performance computing and the architecture of parallel processing systems.

Note: Functionality marked as deferred in this documentation is planned to be implemented in a later release.

Cray XT series supercomputer systems are massively parallel-processing (MPP) systems. Cray has combined commodity and open source components with custom-designed components to create a system that can operate efficiently at immense scale.

Cray XT series systems are based on the Red Storm technology that was developed jointly by Cray Inc. and the U.S. Department of Energy Sandia National Laboratories. Cray XT series systems are designed to run applications that require large-scale processing, high network bandwidth, and complex communications. Typical applications are those that create detailed simulations in both time and space, with complex geometries that involve many different material components. These long-running, resource-intensive applications require a system that is programmable, scalable, reliable, and manageable.

The major features of Cray XT series systems are:

- Cray XT series systems scale from 200 to 30,000 processors. The ability to scale to such proportions stems from the design of system components:
  - The basic scalable component is the *node*. There are two types of nodes. Compute nodes run user applications. Service nodes provide support functions, such as managing the user's environment, handling I/O, and booting the system.

  - Cray XT series systems use a simple memory model. Every image of a distributed application has its own processor and local memory. Remote memory is the memory on the nodes running the associated application images. There is no shared memory.

    Cray XT series systems use error correction code (ECC) technology to detect and correct multiple bit data transfer errors.

    The *system interconnection network* is the data-routing resource that Cray XT series systems use to maintain high communication rates as the number of nodes increases.

- The development environment provides a comprehensive set of compilers, libraries, parallel programming model functions, debuggers, and performance measurement tools.
- The Cray XT series operating system, UNICOS/lc, is tailored to the requirements of computation and service components. Nodes that process user applications run the Catamount microkernel (also known as the *quintessential kernel* or *Qk*). Cray XT series systems support dual-core processing on compute nodes through the Catamount Virtual Node (CVN) capability. A full-featured operating system runs on single-core or dual-core service nodes.
- The Lustre file system scales to thousands of clients and petabytes of data. Lustre can be configured as a parallel file system optimized for large-scale serial access, or as a UFS like file system, optimized for accessing many small files typically seen with root file systems and home directories.
- Reliability, availability, and serviceability (RAS) are designed into system components:
  - Cray XT series system cabinets have only one moving part (a blower that cools the components) and redundant power supplies, reducing the likelihood of cabinet failure.
  - Cray XT series system processor boards (called *blades*) have several reliability features. The blades have redundant voltage regulator modules (VRMs), which are the solid state components most likely to fail. All components are surface-mounted, and the blades have a minimal number of components.
  - All Cray XT series nodes are diskless; the availability of a node is not tied to the availability of a moving part.

  - Most RAID subsystems have multiple redundant RAID controllers with automatic failover capability and multiple Fibre Channel connections to disk storage.

- Cray XT series systems provide advanced system management and system administration features:
  - The Cray XT series single-system view (SSV) is a set of operating system features that provide users and administrators with one view of the system, significantly simplifying the administrator's tasks.
  - The Cray RAS and Management System (CRMS) monitors and manages all major Cray XT series system components. The CRMS is independent of computation and service components and has its own network.
  - Administrators can reboot one or more compute nodes without rebooting the entire Cray XT series system. This feature allows the administrator to reboot compute nodes that were originally booted with Catamount but have stopped running. This reboot capability is also known as *warmboot*.
  - Cray XT series use Ethernet link aggregation (also known as *bonding* or *channel bonding*) to increase aggregate bandwidth by combining multiple Ethernet channels into a single virtual channel. Link aggregation can also be used to increase the availability of a link by using other interfaces in the bond when one of the links in that bond fails.
  - The Cray I/O testing tool provides a common base for testing the I/O components on Cray products. This tool provides a command-line utility, `iotest`, that can be used for high-level diagnostic testing to determine the health of the underlying I/O components in a Cray XT series system.
  - A single physical Cray XT series system can be split into two or more logical machines, each operating as an independent computing resource. A logical machine must have its own compute partition and a service partition that has its own service nodes, external network connections, and I/O equipment. Each logical machine can be booted and dumped independently of the other logical machines. For example, a customer may create a logical machine that has a set of nodes with memory size or processor speed that is different from another logical machine. A job is limited to running within a single logical machine.

A separate boot configuration is provided for each logical machine so that customers can provide separate boot, service database (SDB), and login nodes for each of their logical machines. Once booted, a logical machine will appear as a normal Cray XT series system to the users, limited to the set of hardware included for the logical machine.

The CRMS is common across all logical machines. Because logical machines apply from the system interconnection network layer and up, the CRMS functions continue to behave as a single system for power control, diagnostics, low-level monitoring, etc.

– Support for native IP is provided as an alternative to the default IPPO implementation. Native IP is functionally equivalent to IPPO and has significant performance advantages.

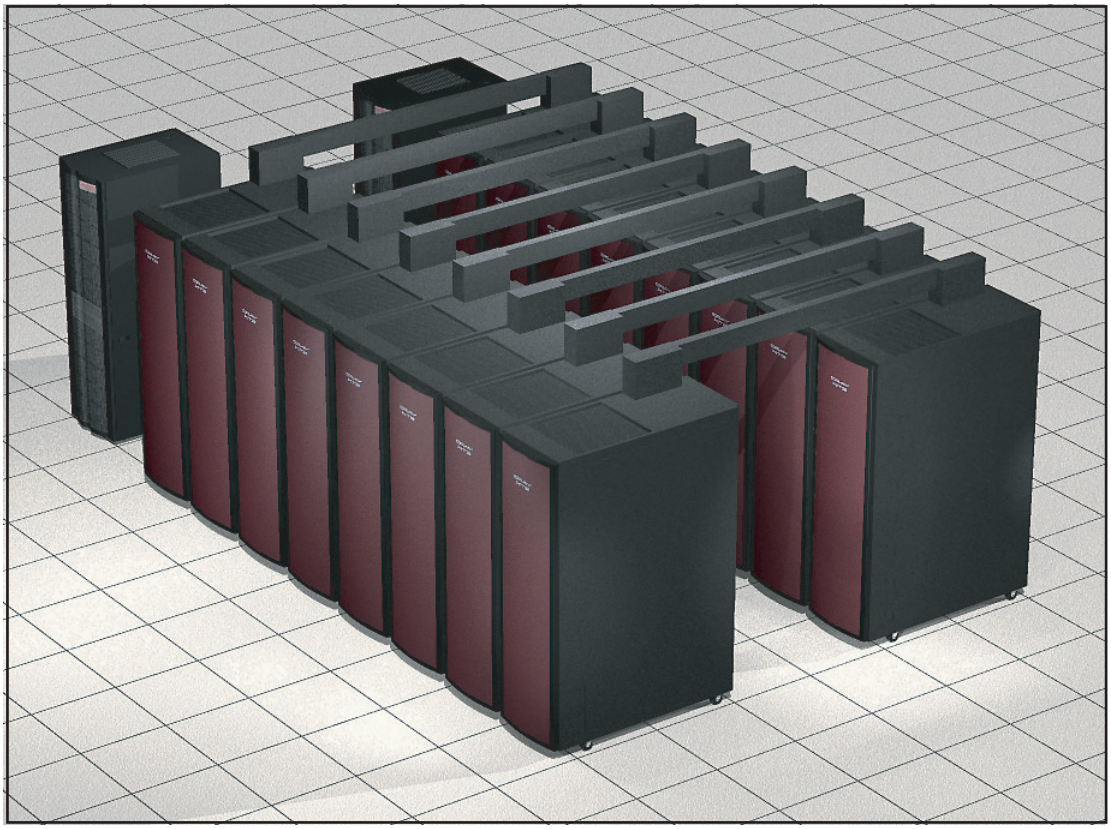

Figure 1. Cray XT Series Supercomputer System

1.1 Related Publications

The Cray XT series system runs with a combination of proprietary, third-party, and open source products, as documented in the following publications.

1.1.1 Publications for Application Developers

- *Cray XT Series System Overview* (this manual)
- *Cray XT Series Programming Environment User's Guide*
- *Cray XT Series Software Release Overview*
- *PGI User's Guide*
- *PGI Tools Guide*
- *PGI Fortran Reference*
- *PGI compiler commands man pages*: `cc`(1), `CC`(1), `ftn`(1), `f77`(1)
- *GCC compiler commands man pages*: `cc`(1), `CC`(1), `f77`(1)
- *GCC manuals*: `http://gcc.gnu.org/onlinedocs/`
- *Modules utility man pages*: `module`(1), `modulefile`(4)
- *Commands related to application launch*: `yod`(1), `xtshowmesh`(1), `xtshowcabs`(1)
- *Cray MPICH2 man pages* (read the `intro_mpi`(1) man page first)
- *Cray SHMEM man pages* (read the `intro_shmem`(1) man page first)
- *LAPACK man pages*
- *ScaLAPACK man pages*
- *BLACS man pages*
- *AMD Core Math Library (ACML)*
- *Cray LibSci FFT man pages* (read the `intro_fft`(3) man page first)
- *FFTW 2.1.5 and 3.1.1 man pages* (read the `intro_fftw2`(3) and / or `intro_fftw3`(3) man page first)
- *SuperLU Users' Guide*

- *PBS Pro 5.3 Quick Start Guide, PBS-3BQ01* [1]
- *PBS Pro 5.3 User Guide, PBS-3BU01* [1]
- *PBS Pro man pages* [1]
- *PBS Pro 5.3 External Reference Specification, PBS-3BE01* [1]
- *PAPI User's Guide*
- *PAPI man pages*
- *PAPI Programmer's Reference*
- *PAPI Software Specification*
- *Using Cray Performance Analysis Tools*
- *CrayPat man pages*: [2] (start with `craypat`(1))
- *Cray Apprentice2 man page* (`app2`(1)) [2]
- *TotalView documentation* [3]
- *GNU debugger documentation* (refer to the `xtgdb`(1)[2] man page and the GDB User Manual at `http://www.gnu.org/software/gdb/documentation/`).
- *UNICOS/lc man pages*
- *SUSE LINUX man pages*
- *Linux documentation* (refer to the Linux Documentation Project at `http://www.tldp.org` and to SUSE documentation at `http://www.suse.com`)

1.1.2 Publications for System Administrators

- *Cray XT Series System Overview* (this manual)
- *Cray XT Series Software Release Overview*
- *Cray XT Series Software Installation and Configuration Guide*

---

1 PBS Pro is an optional product from Altair Grid Technologies available from Cray Inc. Refer to `http://www.altair.com`.

2 An optional Cray product.

3 TotalView is an optional product available from Etnus, LLC (`http://www.etnus.com`).

- *Cray XT Series System Management*
- *Integrating Storage Devices with Cray XT Systems*
- *UNICOS/lc man pages*
- *CRMS man pages*: `xtcli`(8), and `xtgui`(8)
- *SUSE LINUX man pages*
- *Lustre man pages*
- *PBS Pro Release Overview, Installation Guide, and Administration Addendum for Cray XT Series Systems* [4]
- *PBS Pro 5.3 Administrator Guide, PBS-3BA01* [4]
- *Linux documentation* (refer to the Linux Documentation Project at `http://www.tldp.org` and to SUSE documentation at `http://www.suse.com`).

[4] PBS Pro is an optional product from Altair Grid Technologies available from Cray Inc. (refer to `http://www.altair.com`).

Hardware Overview [2]

Hardware for the Cray XT series system comprises computation components, service components, the system interconnection network, and CRMS components. This chapter describes the computation components, service components, and system interconnection network. For a description of CRMS hardware, refer to Chapter 4, page 39.

2.1 Basic Hardware Components

The Cray XT series system comprises the following hardware components:

- AMD Opteron processors
- Dual in-line memory modules (DIMMs)
- Cray SeaStar chips
- System interconnection network

2.1.1 AMD Opteron Processor

The Cray XT series system supports single-core and dual-core AMD Opteron processors. Opteron processors feature:

- Full support of the x86 instruction set.
- Full compatibility with AMD Socket 940 design.
- Out-of-order execution and the ability to issue a maximum of nine instructions simultaneously.
- Sixteen 64-bit registers and a floating-point unit that support full 64-bit IEEE floating-point operations.
- An integer processing unit that performs full 64-bit integer arithmetic.
- Four 48-bit performance counters that can be used to monitor the number or duration of processor events, such as the number of data cache misses or the time it takes to return data from memory after a cache miss.
- A memory controller that uses error correction code (ECC) for memory protection.

Figure 2 shows the components of a single-core processor.

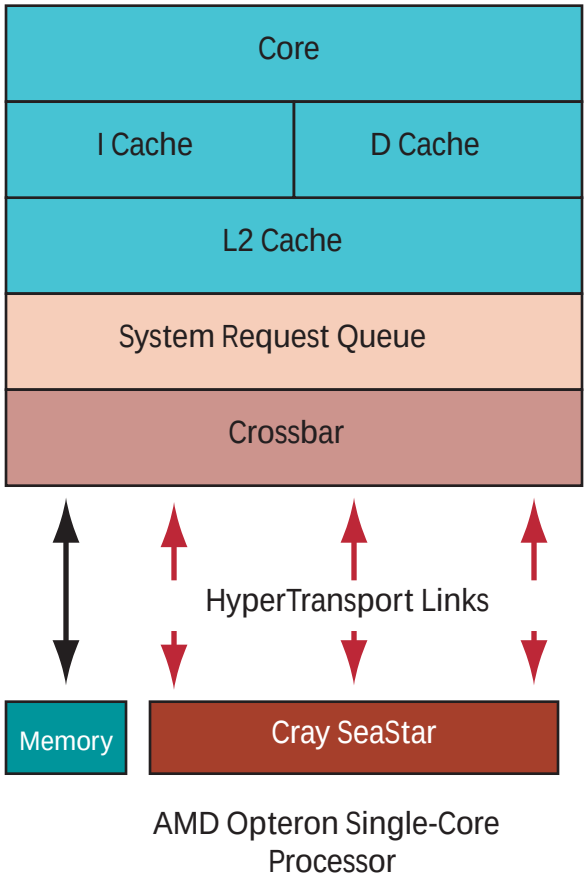

Figure 2. Single-core Processor

Dual-core processors have two computational engines. Each core (also referred to as a *CPU*) has its own execution pipeline and the resources required to run without blocking resources needed by other processes. Because dual-core processor systems can run more tasks simultaneously, they can increase overall system performance. The trade-offs are that each core has less local memory bandwidth (because it is shared by the two cores) and less system interconnection bandwidth (which is also shared).

The Catamount Virtual Node (CVN) capability enables support of dual-core processors on compute nodes.[1]

On service nodes, UNICOS/lc supports dual-core processors by using SUSE LINUX threading capabilities. A dual-core-processor service node functions as a two-way symmetric multiprocessor (SMP).

Figure 3 shows the components of a dual-core processor.

1 CVN was developed by the U.S. Department of Energy Sandia National Laboratories and Cray Inc.

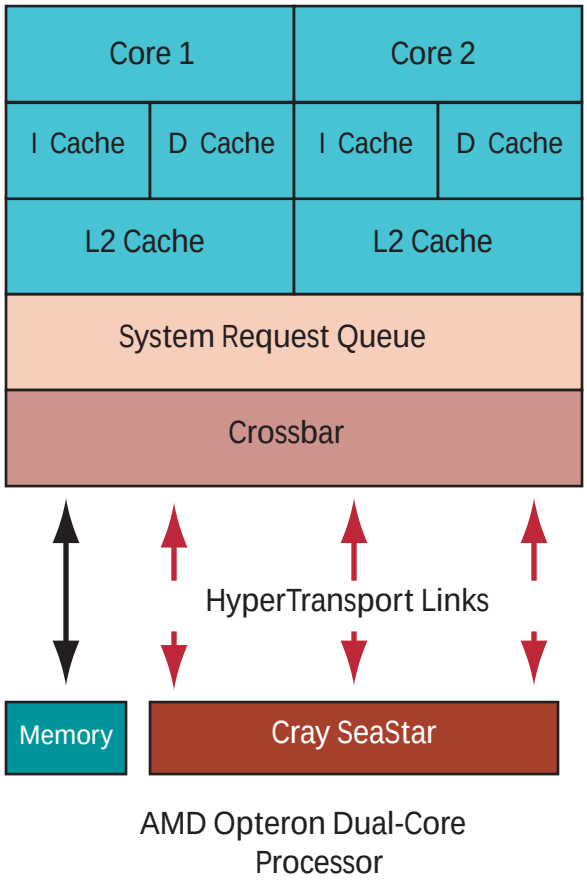

Figure 3. Dual-core Processor

2.1.2 DIMM Memory

The Cray XT series system supports double data rate Dual In-line Memory Modules (DIMMs). DIMM sizes are 512 MB, 1 GB, and 2 GB. With four DIMM slots per node, the maximum physical memory is 8 GB per node. The minimum memory for service nodes is 2 GB.

Cray XT series systems use ECC memory protection technology.

2.1.3 Cray SeaStar Chip

The Cray SeaStar application-specific integrated circuit (ASIC) chip is the system's message processor. The SeaStar chip offloads communications functions from the AMD Opteron processor. A SeaStar chip contains:

- A HyperTransport Link, which connects the AMD Opteron processor to the Cray SeaStar chip.
- A Direct Memory Access (DMA) engine, which manages the movement of data to and from DIMMs. The DMA engine is controlled by an on-board processor.
- A router, which connects the chip to the system interconnection network through high-speed network links. For details, refer to Section 2.1.4, page 14.
- A Portals message passing interface, which provides a data path from an application to memory. Portions of the interface are implemented in Cray SeaStar firmware, which transfers data directly to and from user memory without operating system intervention.
- A link to a blade control processor (also known as the *L0 controller*). Blade control processors are used for booting, monitoring, and maintenance. For more information, refer to Section 4.1.3, page 41.

Figure 4 illustrates the hardware components of the Cray SeaStar chip.

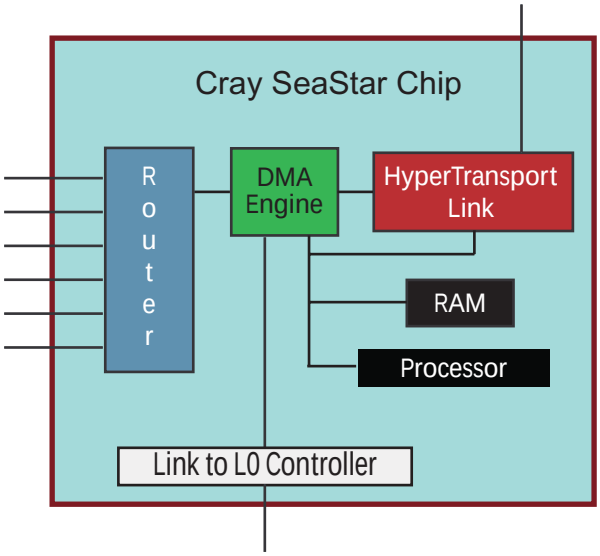

Figure 4. Cray SeaStar Chip

2.1.4 System Interconnection Network

The system interconnection network is the communications center of the Cray XT series system. The network consists of the Cray SeaStar router links and the cables that connect the computation and service nodes.

The network uses a Cray proprietary protocol to provide fast node-to-node message passing and fast I/O to and from a global, shared file system. The network enables the system to achieve an appropriate balance between processor speed and interconnection bandwidth.

2.2 Nodes

Cray XT series processing components combine to form a node. The Cray XT series system has two types of nodes: compute nodes and service nodes. Each node is a logical grouping of a processor, memory, and a data routing resource.

2.2.1 Compute Nodes

Compute nodes run application programs. Each compute node consists of one AMD 940 processor socket with a single-core or dual-core AMD Opteron processor, four DDR1 DIMM slots providing 1 to 8 GB of local memory, and a Cray SeaStar 1 chip that connects the processor to the system interconnection network.

All compute nodes in a logical system use the same processor type. Because processors are inserted into standard AMD Opteron processor sockets, customers can upgrade nodes as faster processors become available. The set of all compute nodes is referred to as the *compute partition*.

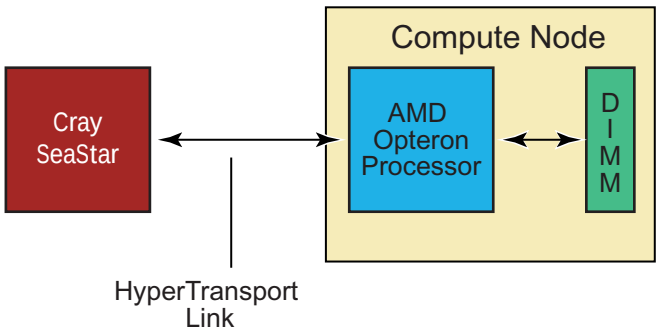

Figure 5. Cray XT Series System Compute Node

2.2.2 Service Nodes

Service nodes handle support functions such as system startup and shutdown, user login, I/O, and network management. Service nodes use single-core or dual-core AMD Opteron processors and SeaStar chips. In addition, each service node has two PCI-X slots. PCI-X cards plug into the slots and interface to external I/O devices. Different PCI-X cards are used for different types of service nodes. The set of all service nodes is referred to as the *service partition*. An administrator-defined portion of a physical Cray XT series system, operating as an independent computing resource, is referred to as a *logical machine*. For a description of the types of service nodes, refer to Section 3.1, page 21.

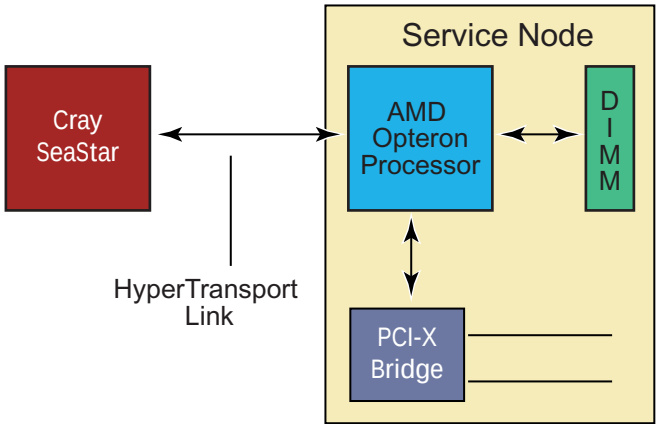

Figure 6. Cray XT Series System Service Node

2.3 Blades, Chassis, and Cabinets

This section describes the main physical components of the Cray XT series system and their configurations.

2.3.1 Blades

While the node is the logical building block of the Cray XT series system, the basic physical component and field-replaceable unit is the *blade*. There are two types of blades: compute blades and service blades.

A compute blade consists of four nodes (one AMD Opteron processor socket, four DIMM slots, and one SeaStar chip per node), voltage regulator modules, and a Cray XT series system L0 controller. Each compute blade within a logical machine is populated with AMD processors of the same type and speed and memory chips of the same speed.

The Cray XT series system L0 controller is a CRMS component; for more information about CRMS hardware, refer to Chapter 4, page 39.

A service blade consists of two nodes, voltage regulator modules, PCI-X cards, and a Cray XT series system L0 controller. Each node contains an AMD Opteron processor socket, four DIMM slots, and a Cray SeaStar chip. A service blade has four SeaStar chips to allow for a common board design and to simplify the interconnect configurations. Several different PCI-X cards are available to provide Fibre Channel, GigE, and 10 GigE interfaces to external devices.

2.3.2 Chassis and Cabinets

Each cabinet contains three vertically stacked chassis, and each chassis contains eight vertically mounted blades. A cabinet can contain compute blades, service blades, or a combination of compute and service blades. A single variable-speed blower in the base of the cabinet cools the components.

A three-phase alternating current (AC) delivers power to the cabinet, where the power is converted to direct current (DC). All cabinets have redundant power supplies. The power supplies and the cabinet control processor (L1 controller) are located at the rear of the cabinet.

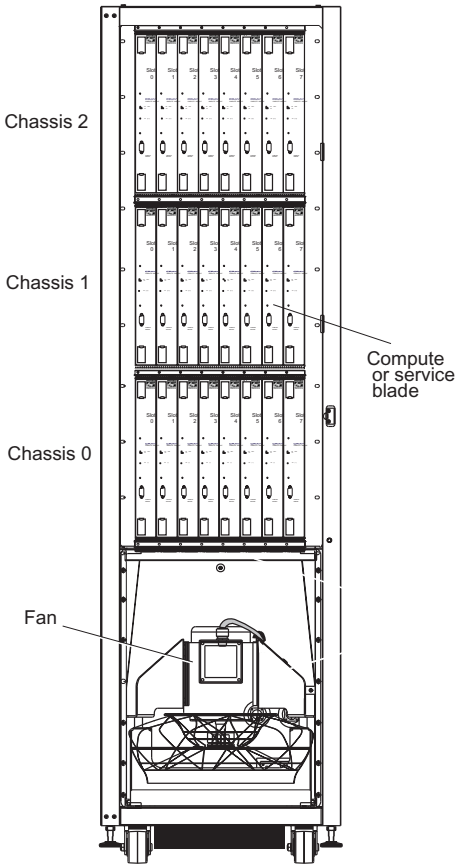

Figure 7. Chassis and Cabinet (front view)

2.4 I/O System

The I/O system handles data transfers to and from storage devices, user applications, and the CRMS. The I/O system consists of service nodes and RAID devices:

- Service nodes provide I/O, network, login, and boot services.
  - I/O service nodes use Fibre Channel PCI-X cards to provide connections to external storage devices.
  - Network service nodes use 10 GigE PCI-X cards to provide connections to network storage devices.
  - Login service nodes use GigE PCI-X cards to provide connections to user workstations.
  - Boot service nodes use Fibre Channel PCI-X cards to provide connections to the boot RAID devices and GigE PCI-X cards to provide connections to the System Management Workstation (SMW). For further information about the SMW, refer to Section 4.1.2, page 41.
- System RAID
- Parallel RAID systems are used for storing user application data, supporting the system boot process, and supporting system management and administration functions.

  RAID storage includes one or more RAID subsystems. A subsystem consists of a singlet or couplet controller and all Fibre Channel or SATA disk enclosures that attach to the controller. Each disk enclosure has 16 disk drive slots with disks arranged in tiers. Each tier consists of data drives and a parity drive. Most RAID components are fully redundant. Fibre Channel disks can be used for the Lustre parallel file system and the system boot RAID. SATA disks are used for the parallel file system.

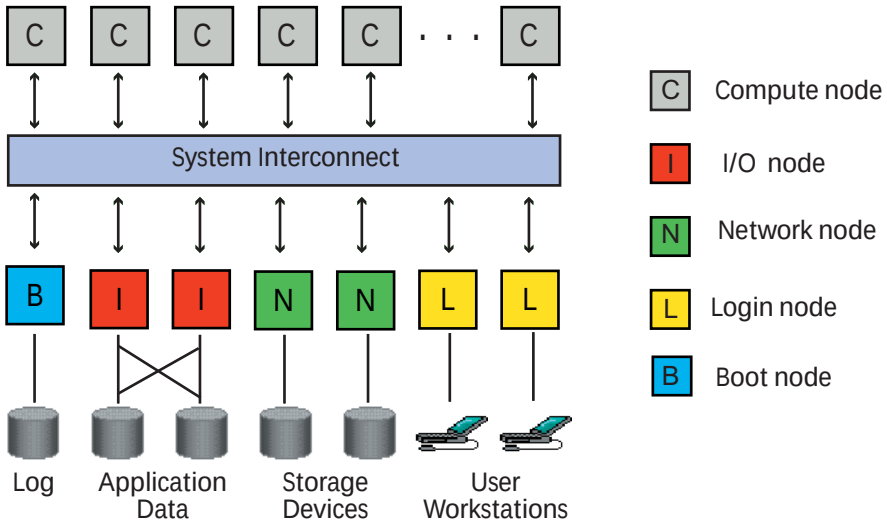

Figure 8. I/O System

Software Overview [3]

Cray XT series systems run a combination of Cray developed software, third-party software, and open source software. The software is optimized for applications that have fine-grain synchronization requirements, large processor counts, and significant communication requirements.

This chapter provides an overview of the UNICOS/lc operating system, the Lustre file system, the application development environment, and system administration tools. For a description of CRMS software, refer to Chapter 4, page 39.

3.1 UNICOS/lc Operating System

The UNICOS/lc operating system consists of service-node and compute-node components.

- Service nodes perform the functions needed to support users, applications, and administrators. Service nodes run a fully featured version of SUSE LINUX. Above the operating system level are specialized daemons and applications that perform functions unique to each service node type.

  There are four basic types of service nodes:

  – *login nodes* on which users log in and PBS Pro resource managers run

  – *I/O nodes* that manage file system metadata and transfer data to and from storage devices and applications

  – *network nodes* that provide Transmission Control Protocol/Internet Protocol (TCP/IP) connections to external systems

  – *system nodes* that perform special services such as system boot, system administration, Service Database (SDB) management (refer to Section 3.4.6, page 37), and syslog management (refer to Section 3.4.5, page 37)

  A system administrator can reconfigure service nodes to improve system efficiency. For more information about configuration options, refer to the *Cray XT Series System Management* and *Integrating Storage Devices with Cray XT Systems* manuals and the *Cray XT Series Software Installation and Configuration Guide*.

- A compute node runs the Catamount microkernel. Sandia National Laboratories developed Catamount to provide support for application execution without the overhead of a full operating system image.

  The microkernel interacts with an application in very limited ways. This minimal interaction enables sites to add compute nodes without a corresponding increase in operating system overhead. It also allows for reproducible run times, a feature of major significance for sites running production systems.

  The microkernel provides virtual memory addressing and physical memory allocation, memory protection, access to the message-passing layer, and a scalable job loader. The microkernel provides limited support for I/O operations. Each instance of a distributed applications is limited to the amount of physical memory of its assigned compute node; the microkernel does not support demand paged virtual memory.

3.2 Lustre File System

I/O nodes host the file system. The Cray XT series system runs Lustre, a high-performance, highly scalable, POSIX-compliant shared file system. Lustre is based on Linux and uses the Portals lightweight message passing API and an object-oriented architecture for storing and retrieving data.

Lustre separates file metadata from data objects. Each instance of a Lustre file system consists of Object Storage Servers (OSSs) and a Metadata Server (MDS). Each OSS hosts two Object Storage Targets (OSTs). Applications use Lustre OSTs to transfer data objects; these data objects can be striped across RAID storage devices.

Lustre's file I/O operations are transparent to the application developer. The I/O functions available to the application developer—Fortran, C, and C++ I/O calls; C and C++ stride I/O calls (`readx`, `writex`, `ireadx`, and `iwritex`); and system I/O calls—are converted to Lustre library calls. Lustre handles file I/O between applications and I/O nodes, and between I/O nodes and RAID storage devices.

For a description of Lustre administration, refer to Section 3.4.3, page 34.

3.3 Cray XT Series Development Environment

The Cray XT series system development environment is the set of software products and services that developers use to build and run applications.

3.3.1 User Environment

The user environment on the Cray XT series system is similar to the environment on a typical Linux workstation. However, there are steps that the user needs to take before starting to work on the Cray XT series system:

1. Set up a secure shell. The Cray XT series system uses `ssh` and `ssh`-enabled applications for secure, password-free remote access to login nodes. Before using `ssh` commands, the user needs to generate an RSA authentication key.
2. Load the appropriate modules. The Cray XT series system uses the Modules utility to support multiple versions of software, such as compilers, and to create integrated software packages. As new versions of the supported software become available, they are added automatically to the Programming Environment, and earlier versions are retained to support legacy applications. By specifying the module to load, the user can choose the default or another version of one or more Programming Environment tools.

For details, refer to the *Cray XT Series Programming Environment User's Guide* and the `module`(1) and `modulefile`(4) man pages.

3.3.2 System Access

To access the Cray XT series system, the user enters the `ssh` command to log in from a standard terminal window.

Logins are distributed among login nodes using a load-leveling service that intercepts login attempts and directs them to the least heavily used login node.

A login node provides all of the standard Linux utilities and commands, a variety of shells, and access to application development tools.

3.3.3 Compilers

The Cray XT series System Programming Environment includes C, C++ , Fortran 90/95, and FORTRAN 77 compilers from The Portland Group (PGI), a wholly owned subsidiary of STMicroelectronics, and the GCC C, C++ , and FORTRAN 77 compilers.

The compilers translate C, C++, and Fortran source programs into Cray XT series system object files. Interlanguage communication functions enable developers to create Fortran programs that call C or C++ routines and C or C++ programs that call Fortran routines.

The command used to invoke a compiler is called a *compilation driver*; it can be used to apply options at the compilation unit level. Fortran directives and C or C++ pragmas apply options to selected portions of code or alter the effects of command-line options.

3.3.3.1 PGI Compiler Commands

The following PGI compiler commands are available:

PGI Compiler	Command
C	`cc`
C++	`CC`
Fortran 90/95	`ftn`
FORTRAN 77	`f77`

For details about PGI compiler command options, refer to the `cc`(1), `CC`(1), `ftn`(1), and `f77`(1) man pages, the *Cray XT Series Programming Environment User's Guide*, and the PGI user documentation.

3.3.3.2 GCC Compiler Commands

The following GCC compiler commands are available:

GCC Compiler	Command
C	`cc`
C++	`c++`
FORTRAN 77	`f77`

For details about GCC compiler command options, refer to the `cc`(1), `CC`(1), and `f77`(1) man pages, and the *Cray XT Series Programming Environment User's Guide*.

3.3.4 Libraries

Developers can use C, C++, and Fortran library functions and the following libraries:

- The system supports the Cray MPICH2 and Cray SHMEM message passing libraries. Message passing through Cray MPICH2 and Cray SHMEM and the underlying Portals layer optimizes communication among applications.

  MPICH2 is an implementation of MPI-2 by the Argonne National Laboratory Group. The Cray XT series Programming Environment includes ROMIO, a high-performance, portable MPI-IO implementation developed by Argonne National Laboratories. The dynamic process (spawn) functions in Cray MPICH2 are not supported at this time, but otherwise the libraries are fully MPI 2.0 compliant.

  Cray SHMEM routines are similar to the Cray MPICH2 routines; they use the Portals low-level message passing layer to pass data between cooperating parallel processes. Cray SHMEM routines can be used in programs that perform computations in separate address spaces and that explicitly pass data to and from different processing elements in the program.

  A new compiler command option, `-default64`, supports the 64-bit module and library for MPI or SHMEM applications using the PGI Fortran 90/95 compiler.

  The Fortran module for Cray SHMEM is not supported. Use the `INCLUDE mpp/shmem.fh` statement instead.

  Portals now supports atomic locking for SHMEM functions. For additional information about SHMEM atomic functions, refer to the *Cray XT Series Programming Environment User's Guide* and the `intro_shmem`(1) man page.

- 64-bit AMD Core Math Library (ACML), which includes:
  - Level 1, 2, and 3 Basic Linear Algebra Subroutines (BLAS)
  - A full suite of Linear Algebra (LAPACK) routines
  - A suite of Fast Fourier Transform (FFT) routines for single-precision, double-precision, single-precision complex, and double-precision complex data types

- Cray XT series LibSci scientific libraries. Cray XT series LibSci contains ScaLAPACK, BLACS, and SuperLU. ScaLAPACK is a set of LAPACK routines redesigned for use in Cray MPICH2 applications. The BLACS package is a set of communication routines used by ScaLAPACK and the user to set up a problem and handle the communications. SuperLU is a set of routines that solve large, sparse, nonsymmetric systems of linear equations. Cray XT series LibSci library routines are written in C but can be called from Fortran, C, or C++ programs.

  The Cray LibSci FFT interfaces are partially supported on Cray XT series systems. Generally, the underlying implementation is through the ACML FFTs. These Cray FFT routines call tuned ACML FFTs. For further information, refer to the *Cray XT Series Programming Environment User's Guide* and the `intro_fft`(3s) man page.

- The Programming Environment includes the 2.1.5 and 3.1.1 releases of FFTW. FFTW is a C subroutine library with Fortran interfaces for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, such as the discrete cosine/sine transforms or DCT/DST). The fast Fourier transform (FFT) algorithm is applied for many problem sizes. Distributed memory parallel (DMP) FFTs are available only in FFTW 2.1.5. For further information, refer to the *Cray XT Series Programming Environment User's Guide* and the `intro_fftw2`(3) and `intro_fftw3`(3) man pages.

- A subset of the GNU C Language Runtime Library, glibc. For details, refer to the *Cray XT Series Programming Environment User's Guide*.

- IOBUF is an optional I/O buffering library that can reduce the I/O wait time for programs that read or write large files sequentially. IOBUF intercepts standard I/O calls, such as read and open, and replaces the stdio (glibc, libio) layer of buffering with an additional layer of buffering, thus improving program performance by enabling asynchronous prefetching and caching of file data. IOBUF can also gather runtime statistics and print a summary report of I/O activity for each file.

3.3.5 Linker

After correcting compilation errors, the developer again invokes the compilation driver, this time specifying the application's object files (*filename*`.o`) and libraries (*filename*`.a`) as required.

The linker extracts the library modules that the program requires and creates the executable file (named `a.out` by default).

3.3.6 Runtime Environment

There are two methods of running applications: as interactive jobs and as batch jobs. There are two types of interactive jobs:

- Jobs launched directly through `yod` commands
- PBS Pro interactive jobs launched through the `qsub -I` and `yod` commands

Both single-program-multiple-data (SPMD) and multiple-program-multiple-data (MPMD) applications are supported.

Before launching a job, the user can run the `xtshowmesh` or `xtshowcabs` utility to view the current state of the system. This utility displays the status of the compute nodes—whether they are up or down, designated for interactive or batch processing, and free or in use. For more information, refer to the `xtshowmesh`(1) and `xtshowcabs`(1) man pages and the *Cray XT Series Programming Environment User's Guide*.

Some runtime components must run on a service node. To facilitate this, the `yod` utility executes on a service node on behalf of the application.

3.3.6.1 Interactive Jobs

To run an interactive job, the user enters the `yod` command, specifying all executables that are part of the job and the number of nodes on which they will run. The user can request a number of compute nodes or, using `xtshowmesh` or `xtshowmesh` data, a list of specific compute nodes.

The `yod` utility queries the Compute Processor Allocator (CPA) to get a list of nodes on which to run the job. It then contacts the process control thread (PCT) for the first node on the list. After connecting with the PCT, `yod` maps the executables, one at a time, into its address space and sets up a memory region for the PCT to fetch the executable.

In response to the load request, the PCT propagates the executable to all compute nodes selected to run that program. After the first PCT has finished loading the executable, it maps its memory for load propagation, contacts the next set of PCTs, and forwards the load request. This process is repeated until all compute nodes have loaded. After a compute node and all of its descendants have finished loading, the PCT starts the executable.

On dual-core processor systems, there is only one instance of the microkernel and the PCT per node. The PCT runs only on the "master" CPU, though parts of QK do run on the "subordinate" CPU. The PCT is able to schedule user processes on either processor.

While the application is running, yod provides I/O services, propagates signals, and participates in cleanup when the application terminates.

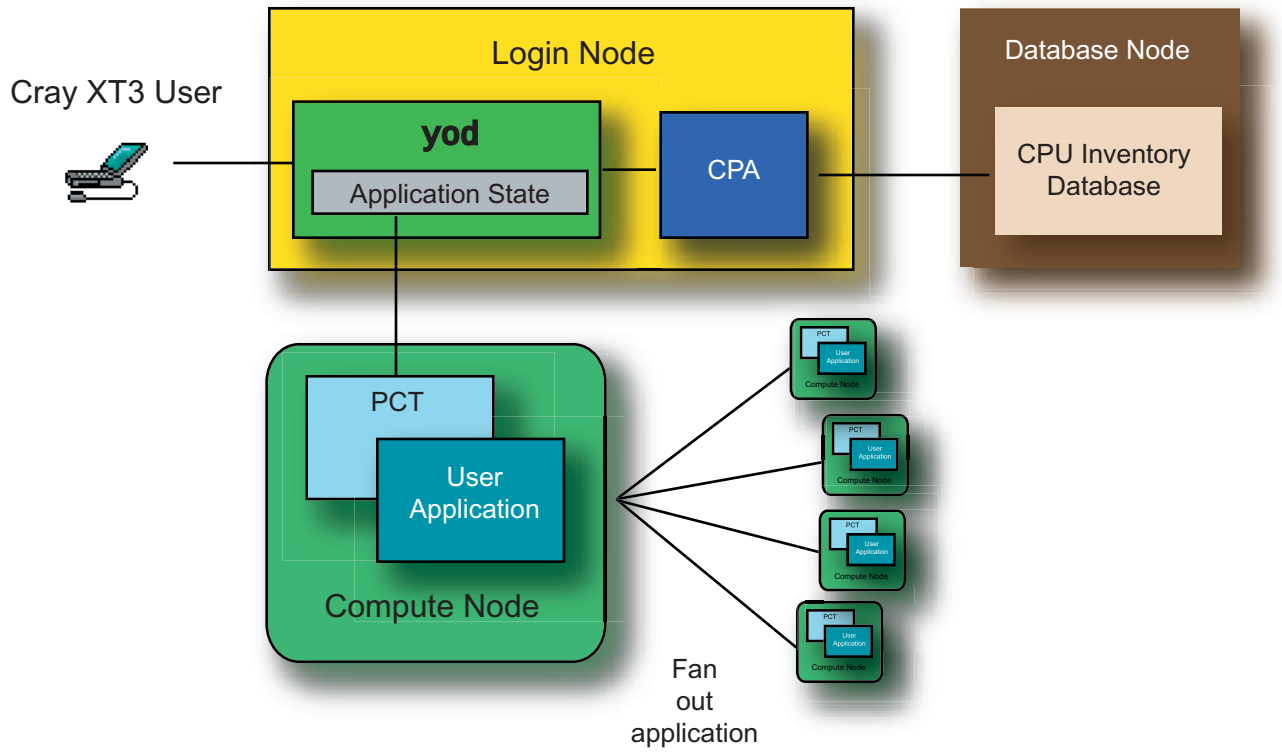

Figure 9. Launching Interactive Jobs

3.3.6.2 Batch Jobs

The Cray XT series system uses PBS Pro to launch batch jobs. PBS Pro is a networked subsystem for submitting, monitoring, and controlling a workload of batch jobs.

A batch job is typically a shell script and a set of attributes that provide resource and control information about the job. Batch jobs are scheduled for execution at a time chosen by the subsystem according to a defined policy and the availability of resources.

The user logs on to the Cray XT series system and creates a script containing the yod command(s) to run the application. The user then enters the PBS Pro qsub command to submit the job to a PBS Pro server.

A PBS Pro server, executing on a login node, maintains a set of job queues. Each queue holds a set of jobs available for execution and each job has a set of user-specified resource requirements.

The user can specify the number of compute nodes to allocate to the job. If no number is given, the job is allocated one node.

The PBS Pro scheduler is the policy engine that examines the set of ready-to-run jobs and selects the next job to run based on a set of criteria. The PBS Pro scheduler negotiates with the CPA to allocate the compute node or nodes on which the job is to run. PBS Pro communicates with a PBS Pro daemon running on the login node or a dedicated batch support node. As with interactive jobs, the yod utility queries the CPA to get the list of compute nodes allocated to the job, then requests the PCT to fetch the executable and propagate it to all compute nodes selected to run that program.

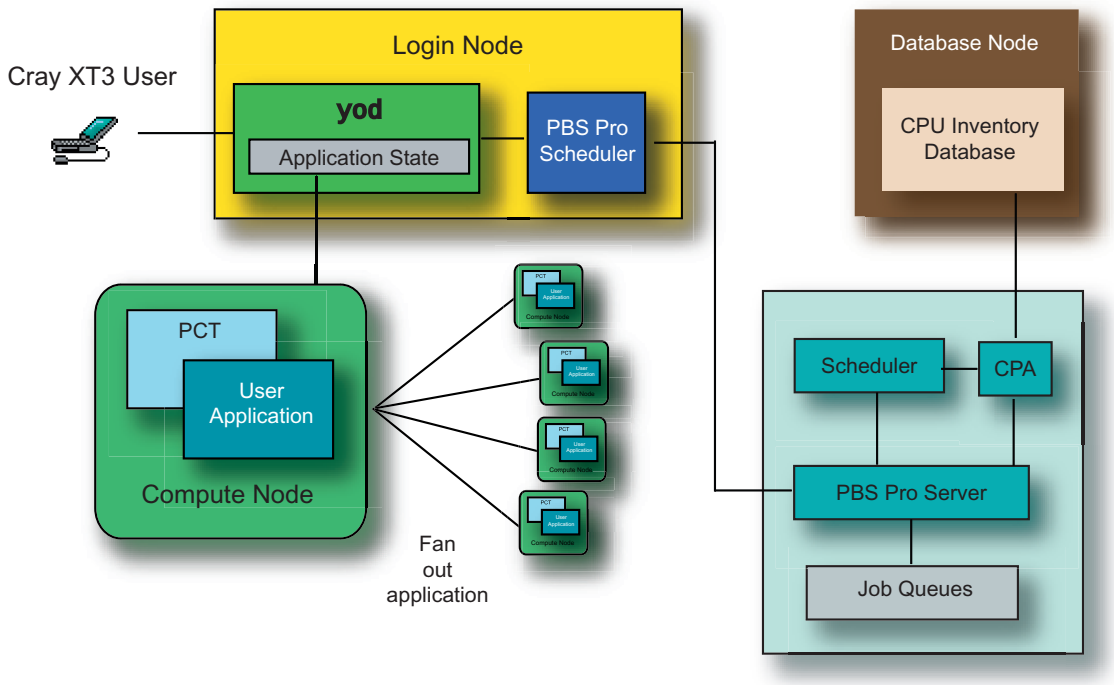

Figure 10. Launching Batch Jobs

3.3.7 Debugging Applications

The Cray XT series system supports the Etnus TotalView debugger for single-process and mutiprocess debugging and the GNU Project debugger, gdb, for single-process debugging.

A customized implementation of the Etnus TotalView debugger, available from Etnus, LLC. provides source-level debugging of applications and is compatible with the PGI C, C++, and Fortran compilers and the GCC C, C++, and FORTRAN 77 compilers.

TotalView can debug applications running on 1 to 1024 compute nodes, providing visibility and control into each process individually or by groups. It supports access to MPI-specific data, such as the message queues.

To debug a program using TotalView, the developer invokes TotalView with the `totalview` command. TotalView parses the command line to get the number of nodes, then makes a node allocation request to the CPA. TotalView directs `yod` to load but not start the application. The `yod` utility then loads the application onto the compute nodes, after which TotalView can perform initial setup before instructing `yod` to start the application.

The developer can use gdb to debug a GCC C, C++, or FORTRAN 77 single-process program. The developer invokes the GNU debugger with the `xtgdb` command.

For more information about TotalView, refer to the *Cray XT Series Programming Environment User's Guide* and Etnus TotalView documentation (Section 1.1.1, page 5).

For more information about gdb, refer to the *Cray XT Series Programming Environment User's Guide*, the `xtgdb`(1) man page, and gdb documentation (Section 1.1.1, page 5).

3.3.8 Measuring Performance

The Cray XT series system provides tools for the collection, display, and analysis of performance data.

3.3.8.1 Performance API (PAPI)

The Performance API (PAPI) from the University of Tennessee and Oak Ridge National Laboratory is a standard interface for access to hardware performance counters. A PAPI event set maps AMD Opteron processor hardware counters to a list of events such as Level 1 data cache misses, data translation lookaside buffer (TLB) misses, and cycles stalled waiting for memory accesses. Developers can use the API to collect data on those events.

3.3.8.2 CrayPat

CrayPat is a performance analysis tool. It is an optional product available from Cray Inc. The developer can use CrayPat to perform trace experiments on an instrumented application and analyze the results of those experiments. Trace experiments keep track of the number of specific events that occur, for example, the number of times a particular system call is executed.

The developer uses the `pat_build` command to instrument programs. No recompilation is needed to produce the instrumented program.

After instrumenting a program, the developer sets environment variables to control run time data collection, runs the instrumented program, then uses the `pat_report` command to generate a report.

3.3.8.3 Cray Apprentice2

Cray Apprentice2 is an interactive X Window System tool for displaying data captured by CrayPat during program execution.

Cray Apprentice2 identifies the following conditions:

- Load imbalance
- Excessive serialization
- Excessive communication
- Network contention
- Poor use of the memory hierarchy
- Poor functional unit use

Cray Apprentice2 has the following capabilities:

- It is a post-execution performance analysis tool that provides information about a program by examining data files that were created during program execution. It is not a debugger or a simulator.
- Cray Apprentice2 displays two types of performance data: call stack sampling and Cray MPICH2 tracing, contingent on the data that was captured by CrayPat during execution.
- It reports time statistics for all processing elements and for individual routines.
- It shows total execution time, synchronization time, time to execute a subroutine, communication time, and the number of calls to a subroutine.

3.4 System Administration

The system administration environment provides the tools that administrators use to manage system functions, view and modify the system state, and maintain system configuration files. System administration components are a combination of Cray XT series system hardware, SUSE LINUX, Lustre, and Cray XT series system-specific utilities and resources.

Note: For details about standard SUSE LINUX administration, refer to `http://www.tldp.org` or `http://www.suse.com`. For details about Lustre functions, refer to the *Cray XT Series System Management* manual and `http://www.lustre.org/documentation.html`.

Many of the components used for system administration are also used for CRMS (such as powering up and down and monitoring the health of hardware components). For details about CRMS, refer to Chapter 4, page 39.

3.4.1 System Management Workstation (SMW)

The SMW is a server and display that provides a single-point interface to an administrator's environment. The SMW provides a terminal window from which the administrator performs tasks like adding user accounts, changing passwords, and monitoring applications. The SMW accesses system components through the administration / CRMS network; it does not use the system interconnection network.

3.4.2 Shared-Root File System

The Cray XT series system has a shared-root file system where the root directory is shared read-only on the service nodes. All nodes have the same default directory structure. However, the `/etc` directory is specially mounted on each service node as a node-specific directory of symbolic links. The administrator can change the symbolic links in the `/etc` directory by the process of specialization, which changes the symbolic link to point to a non-default version of a file. The administrator can specialize files for individual nodes or for a class (type) of nodes.

The administrator's interface includes commands to view file layout from a specified node, determine the type of specialization, and create a directory structure for a new node or class based on an existing node or class. For details, refer to the *Cray XT Series System Management* manual.

3.4.3 Lustre File System Administration

The administrator can configure the Lustre file system to optimize I/O for a broad spectrum of application requirements. At one extreme is Lustre as a UFS-like file system. Configuring Lustre as a UFS-like file system optimizes it for access patterns of directories with many small files such as is typically seen with root file systems and often with home directories. At the other extreme is Lustre as a parallel file system. Configuring Lustre as a parallel file system optimizes it for large-scale, serial access typical of many Cray MPICH2 and Cray SHMEM applications.

When a file is created, the client contacts a metadata server (MDS), which creates an inode for the file. The inode holds metadata rather than references to the actual data. The MDS handles namespace operations, such as opening or closing a file, managing directory listings, and changing permissions. The MDS contacts Object Storage Targets (OSTs) to create data objects. The OSTs handle block allocation, enforce security for client access, and perform parallel I/O operations to transfer file data. The administrator can create and mount more than one instance of Lustre. One MDS plus one or more OSTs make up a single instance of Lustre and are managed together.

Objects allocated on OSTs hold the data associated with the file. Once a file is created, read and write operations take place directly between the client and the OST, bypassing the MDS. The OSTs use Linux ext3 file systems for backend storage. These file systems are used to store Lustre file and metadata objects and are not directly visible to the user. Lustre does not support building a parallel file system with NFS file systems.

Lustre OST/MDS manual failover provides a service that switches to a standby server when the primary system fails or the service is temporarily shut down for maintenance. For Lustre file systems on Cray XT series systems, failover is not automatic but can be done manually if the system is set up to do so. For a Linux client, manual failover is completely transparent to an application. For a Catamount client, only applications started after the failover will be able to use the failover service.

Lustre configuration information is maintained in the SDB. For details, refer to the *Cray XT Series System Management* manual and the *Cray XT Series Software Installation and Configuration Guide*.

3.4.3.1 Lustre UFS-Like File System

Lustre supports a version of UFS that features serial access to files and a uniform file space with full POSIX compliance. A single metadata server and a single OST provide the paths for data transfers between applications and RAID devices.

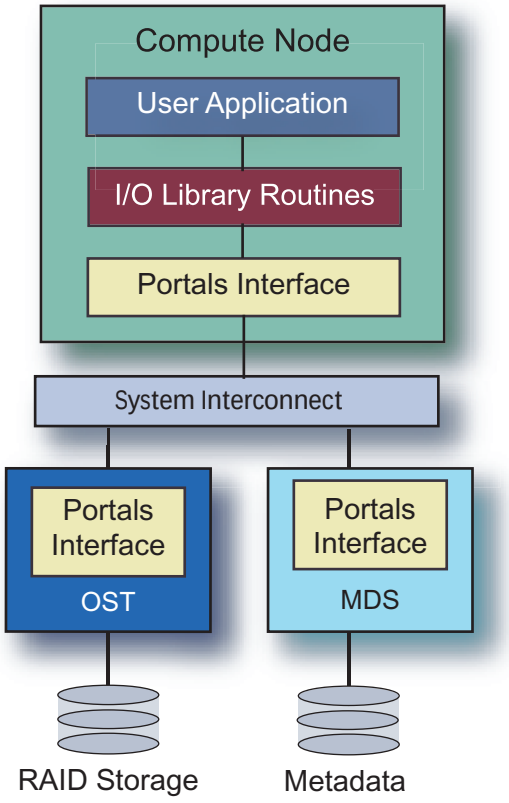

Figure 11. Lustre UFS-Like File System

3.4.3.2 Lustre Parallel File System

The administrator can configure Lustre as a parallel file system by creating multiple object storage targets (OSTs). The file system optimizes I/O by striping files across many RAID storage devices.

The administrator can specify a default system-wide striping pattern at file system creation time. The administrator specifies:

- The default number of bytes stored on each OST
- The default number of OSTs across which each file is striped

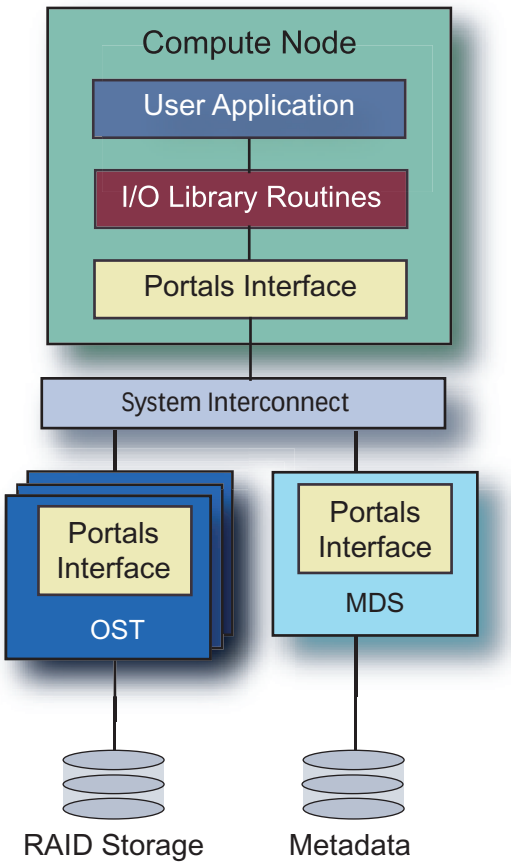

Figure 12. Lustre Parallel File System

3.4.4 Configuration and Source Files

The administrator uses boot nodes to view files, maintain configuration files, and manage the processes of executing programs. Boot nodes connect to the SMW and are accessible through a login shell.

The `xtopview` utility runs on boot nodes and allows the administrator to view files as they would appear on any node. The administrator uses this tool to coordinate changes to configuration files and software source files. All operations are logged.

The `xtopview` utility maintains a database of files to monitor as well as file state information such as checksum and modification dates. Messages about file changes are saved through a Revision Control System (RCS) utility.

3.4.5 System Log

Once the system is booted, console messages are sent to the system log and are written to the boot RAID system. System log messages generated by service node kernels and daemons are gathered by syslog daemons running on all service nodes. Kernel errors and panic messages are sent directly to the SMW.

The administrator can configure the syslog daemon to write the messages to different files, sorted by message generator or degree of importance.

3.4.6 Service Database (SDB)

A database node hosts the Service Database (SDB), which is accessible from every service processor.

The SDB, implemented in MySQL, contains the following information:

- Global state information for the CPA. CPA daemons synchronize resource allocation and store their persistent state in database tables in the Service Database. The CPA interacts with the system accounting utilities to keep track of free and allocated jobs. It logs job parameters such as start time, nodes allocated, and error conditions.
- Accounting information (processor usage, disk storage usage, file I/O demand, application execution time, etc.). The information is accessed by accounting software that accumulates resource usage information, system availability, and system utilization and generates system accounting reports.
- System configuration tables that list and describe the configuration files.

Cray XT series system accounting is a set of utilities that run on a service node. Accounting information is collected in the Service Database (SDB). The administrator uses accounting commands to generate the following reports from the SDB:

- Job and system use
- All jobs in a time period
- System-wide statistics
- Raw accounting data
- System accounting error messages

3.4.7 System Activity Reports

The `sar`(1) command collects, reports, or saves system activity information for service nodes. To get system activity information (such as accounting information) for compute nodes, use the `xtgenacct` command instead. For more information, refer to the `sar`(1) and `xtgenacct`(8) man pages.

Cray RAS and Management System (CRMS) [4]

The Cray RAS and Management System (CRMS) is an integrated, independent system of hardware and software that monitors Cray XT series system components, manages hardware and software failures, controls startup and shutdown processes, manages the system interconnection network, and displays the system state to the administrator. The CRMS interfaces with all major hardware and software components of the system.

Because the CRMS is a completely separate system with its own processors and network, the services that it provides do not take resources from running applications. In addition, if a component fails, the CRMS continues to provide fault identification and recovery services and enables the functioning parts of the system to continue operating.

For detailed information about the CRMS, refer to the *Cray XT Series System Management* manual.

4.1 CRMS Hardware

The hardware components of CRMS are the CRMS network, the System Management Workstation (SMW), the blade control processors (L0 controllers), and the cabinet control processors (L1 controllers). CRMS hardware monitors compute and service node components, operating system heartbeats, power supplies, cooling fans, voltage regulators, and RAID systems.

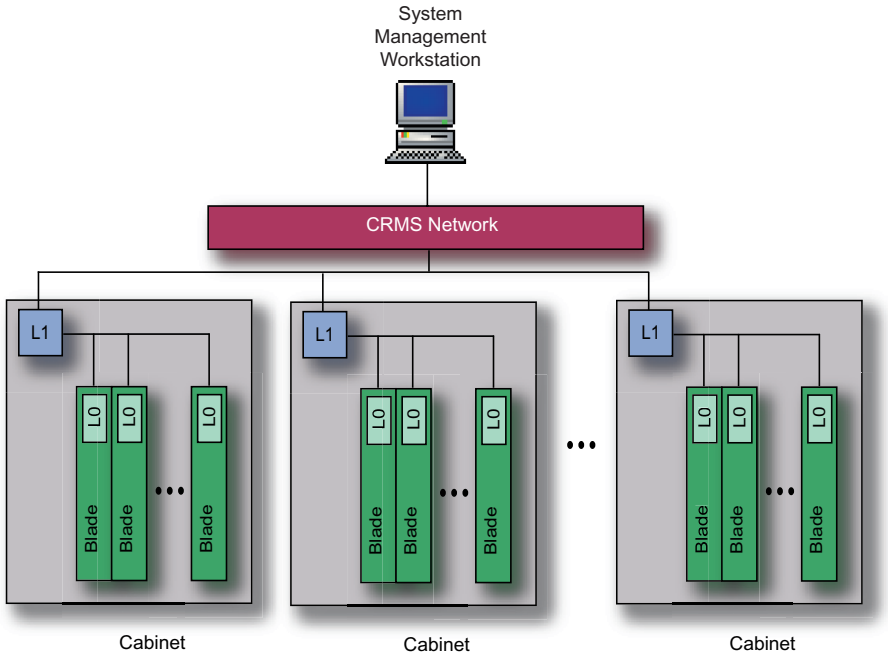

Figure 13. CRMS Components

4.1.1 CRMS Network

The CRMS network is an Ethernet connection between the SMW and the components that the CRMS monitors.

The CRMS network's function is to provide an efficient means of collecting status from and broadcasting messages to system components. The CRMS network is separate from the system interconnection network.

Traffic on the CRMS network is normally low, with occasional peaks of activity when major events occur. There is a baseline level of traffic to and from the hardware controllers. All other traffic is driven by events, either those due to hardware or software failures or those initiated by the administrator. The highest level of network traffic occurs during the initial booting of the entire system as console messages from the booting images are transmitted onto the network. This level of traffic is well within the capacity of the network.

4.1.2 System Management Workstation

The system management workstation (SMW) is the administrator's single-point interface for booting, monitoring, and managing Cray XT series system components. The SMW consists of a server and a display device. Multiple administrators can use the SMW, locally or remotely over an internal LAN or WAN.

Note: The SMW is also used to perform system administration functions (refer to Section 3.4, page 33).

4.1.3 Hardware Controllers

At the lowest level of the CRMS are the L0 and L1 controllers that monitor the hardware and software of the components on which they reside.

Every compute blade and service blade has a blade control processor (L0 controller) that monitors the components on the blade, checking status registers of the AMD Opteron processors, the Control Status Registers (CSRs) of the Cray SeaStar chip, and the voltage regulation modules (VRMs). The L0 controllers also monitor board temperatures and the UNICOS/lc heartbeat.

Each cabinet has a cabinet control processor (L1 controller) that communicates with the L0 controllers within the cabinet and monitors the power supplies and the temperatures of the air cooling the blades. Each L1 controller also routes messages between the L0 controllers in its cabinet and the SMW.

4.2 CRMS Software

The CRMS software consists of software monitors; the administrator's CRMS interfaces; and event probes, loggers, and handlers. This section describes the software monitors and administrator interfaces. For a description of event probes, loggers, and handlers, refer to Section 4.3, page 43.

4.2.1 Software Monitors

Resiliency communication agents (RCAs) run on all compute nodes and service nodes. RCAs are the primary communications interface between the node's operating environment and the CRMS components external to the node. They monitor software services and the operating system instance on each node.

Each RCA provides an interface between the CRMS and the system processes running on a node for event notification, informational messages, information requests, and probing. The RCA also provides a subscription service for processes running on the nodes. This service notifies the current node of events on other nodes that may affect the current node or that require action by the current node or its functions.

Each RCA generates a periodic heartbeat message so that the CRMS can know when an RCA has failed. Failure of an RCA heartbeat is interpreted as a failure of the UNICOS/lc operating system on that node.

RCA daemons running on each node start a CRMS process called *failover manager*. If a service fails, the RCA daemon broadcasts a service-failed message to the CRMS.

Failover managers on other nodes register to receive these messages. Each failover manager checks to determine if it is the backup for any failed services that relate to the message and, if it is, directs the RCA daemon on its node to restart the failed service.

4.2.2 CRMS Administrator Interfaces

The CRMS provides both a command-line and a graphical interface. The `xtcli` command is the command line interface for managing the Cray XT series system from the SMW. The `xtgui` command launches the graphical interface. In general, the administrator can perform any `xtcli` function with `xtgui` except boot.

The SMW is used to monitor data, view status reports, and execute system control functions. If any component of the system detects an error, it sends a message to the SMW. The message is logged and displayed for the administrator. CRMS policy decisions determine how the fault is handled. The SMW logs all information it receives from the system to a RAID storage device to ensure the information is not lost due to component failures.

In the event of an SMW problem, SMW failover provides:

- SMW scheduled shutdown and reboot of the same SMW.
- SMW unscheduled shutdown and reboot of the same SMW.
- SMW scheduled shutdown and startup on a second SMW.
- SMW unscheduled shutdown and startup on another SMW.

4.3 CRMS Actions

The CRMS manages the startup and shutdown processes and event probing, logging, and handling.

The CRMS collects data about the system (event probing and logging) that is then used to determine which components have failed and in what manner. After determining that a component has failed, the CRMS initiates some actions (event handling) in response to detected failures that, if left unattended, could cause worse failures. The CRMS also initiates actions to prevent failed components from interfering with the operations of other components.

4.3.1 System Startup and Shutdown

The administrator starts a Cray XT series system by powering up the system, booting the software on the compute nodes and service nodes, adding the booted nodes to the system interconnection network, starting the RCA daemons, and starting the compute processor allocator (CPA). The administrator stops the system by reserving, removing, and stopping components and powering off the system. For logical machines, the administrator can boot, run diagnostics, run user applications, and power down without interfering with other logical machines as long as the CRMS is running on the SMW and the machines have separate file systems.

For details about the startup and shutdown processes, refer to the *Cray XT Series System Management* manual and the `xtcli`(8) man page.

4.3.2 Event Probing

The CRMS probes are the primary means of monitoring hardware and software components of a Cray XT series system. The CRMS probes that are hosted on the SMW collect data from CRMS probes running on the L0 and L1 controllers and RCA daemons running on the compute nodes. In addition to dynamic probing, the CRMS provides an offline diagnostic suite that probes all CRMS-controlled components.

4.3.3 Event Logging

The event logger preserves data that the administrator uses to determine the reason for reduced system availability. It runs on the SMW and logs all status and event data generated by:

- CRMS probes
- Processes communicating through RCA daemons on compute and service nodes
- Other CRMS processes running on L0 and L1 controllers

Event messages are time stamped and logged. Before the message reaches the input queue of the event handler, an attempt is made to recover from a failure. If a compute or service blade fails, the CRMS notifies the administrator. (Deferred implementation) The administrator can hot swap the blade without affecting other jobs in the system or the system as a whole. After blade replacement, the administrator reintroduces those processors into the system.

4.3.4 Event Handling

The event handler evaluates messages from CRMS probes and determines what to do about them. The CRMS is designed to prevent single-point failures of either hardware or system software from interrupting the system. Examples of single-point failures that are handled by the CRMS system are:

- Compute node failure. A failing compute node is automatically isolated and shut down. The failure affects only the application running on that node; the rest of the system continues running and servicing other applications.
- Power supply failure. Power supplies have an N+1 configuration for each chassis in a cabinet; failure of an individual power supply does not cause an interrupt of a compute node.

In addition, the CRMS broadcasts failure events over the CRMS network so that each component can make a local decision about how to deal with the fault. For example, both the L0 and L1 controllers contain code to react to critical faults without administrator intervention.

Glossary

blade

1) A field-replaceable physical entity. A service blade consists of AMD Opteron sockets, memory, Cray SeaStar chips, PCI-X cards, and a blade control processor. A compute blade consists of AMD Opteron sockets, memory, Cray SeaStar chips, and a blade control processor. 2) From a system management perspective, a logical grouping of nodes and blade control processor that monitors the nodes on that blade.

blade control processor

A microprocessor on a blade that communicates with a cabinet control processor through the CRMS network to monitor and control the nodes on the blade. See also *blade*; *Cray RAS and Management System (CRMS).*

cabinet control processor

A microprocessor in the cabinet that communicates with the CRMS through the CRMS network to monitor and control the devices in a system cabinet. See also *Cray RAS and Management System (CRMS).*

cage

A chassis on a Cray XT series system. See *chassis.*

Catamount

The microkernel operating system developed by Sandia National Laboratories and implemented to run on Cray XT series single-core compute nodes. See also *Catamount Virtual Node (CVN), compute node.*

Catamount Virtual Node (CVN)

The Catamount microkernel operating system enhanced to run on dual-core Cray XT series compute nodes.

chassis

The hardware component of a Cray XT series cabinet that houses blades. Each cabinet contains three vertically stacked chassis, and each chassis contains eight vertically mounted blades. See also *cage.*

compute blade

See *blade*.

compute node

Runs a microkernel and performs only computation. System services cannot run on compute nodes. See also *node*; *service node*.

compute partition

The logical group that consists of all compute nodes.

compute processor allocator (CPA)

A program that coordinates with yod to allocate processing elements.

Cray RAS and Management System (CRMS)

A system of software and hardware that implements reliability, availability, and serviceability (RAS) and some system management functions. The CRMS components use a private Ethernet network, not the system interconnection network. See also *system interconnection network*.

Cray SeaStar chip

The component of the system interconnection network that provides message routing and communication services. See also *system interconnection network*.

CrayDoc

Cray's documentation system for accessing and searching Cray books, man pages, and glossary terms from a Web browser.

deferred implementation

The label used to introduce information about a feature that will not be implemented until a later release.

distributed memory

The kind of memory in a parallel processor where each processor has fast access to its own local memory and where to access another processor's memory it must send a message through the interprocessor network.

dual-core processor

A processor that combines two independent execution engines ("cores"), each with its own cache and cache controller, on a single chip.

L0 controller

See *blade control processor*.

L1 controller

See *cabinet control processor*.

logical machine

An administrator-defined portion of a physical Cray XT series system, operating as an independent computing resource.

login node

The service node that provides a user interface and services for compiling and running applications.

metadata server (MDS)

The component of the Lustre file system that stores file metadata.

module

See *blade*.

Modules

A package on a Cray system that enables you to dynamically modify your user environment by using module files. (This term is not related to the module statement of the Fortran language; it is related to setting up the Cray system environment.) The user interface to this package is the `module` command, which provides a number of capabilities to the user, including loading a module file, unloading a module file, listing which module files are loaded, determining which module files are available, and other such capabilities.

node

For UNICOS/lc systems, the logical group of processor(s), memory, and network components acting as a network end point on the system interconnection network. See also *processing element*.

object storage target (OST)

The component of the Lustre file system that handles file activities.

parallel processing

Processing in which multiple processors work on a single application simultaneously.

Portals

A message-passing interface that enables scalable, high-performance network communication between nodes. Applications communicating at the user level link to the Cray MPICH2 or Cray SHMEM library. The Portals interface is transparent to the application programmer.

processing element

The smallest physical compute group. There are two types of processing elements: a compute processing element consists of an AMD Opteron processor, memory, and a link to a Cray SeaStar chip. A service processing element consists of an AMD Opteron processor, memory, a link to a Cray SeaStar chip, and PCI-X links.

reliability, availability, serviceability (RAS)

System hardware and software design that achieves increased system availability by avoiding or recovering from component and system failures.

resiliency communication agent (RCA)

A communications interface between the operating environment and the CRMS. Each RCA provides an interface between the CRMS and the processes running on a node and supports event notification, informational messages, information requests, and probes. See also *Cray RAS and Management System (CRMS)*.

service blade

See *blade*.

service database (SDB)

The database that maintains the global system state.

service node

A node that performs support functions for applications and system services. Service nodes run SUSE LINUX and perform specialized functions. There are six types of predefined service nodes: login, IO, network, boot, database, and syslog.

service partition

The logical group of all service nodes.

specialization

The process of setting files on the shared-root file system so that unique files can be present for a node or for a class of node.

system interconnection network

The high-speed network that handles all node-to-node data transfers.

system management workstation (SMW)

The workstation that is the single point of control for the CRMS and Cray XT3 system administration. See also *Cray RAS and Management System (CRMS).*

UNICOS/lc

The operating system for Cray XT series systems.

Index

10 GigE interfaces, 16

A

Accessing the system, 23
Accounting, 37–38
ACML, 25
Administration, 33, 41
Application development tools, 2
Application launch, 27
Authentication, 23

B

Batch jobs, 27–28
BLACS, 26
Blade, 2, 16
  compute, 16
  service, 16
Blade control processor, 13, 39, 41
Board (processor)
  See blade
Bonding
  See Ethernet link aggregation
Boot nodes, 21, 37
Boot service nodes, 18

C

C and C++ interlanguage communications, 23
C command, 23
C++ command, 23
Cabinet, 16–17
Cabinet control processor, 39, 41
Catamount, 22
Catamount Virtual Node (CVN), 2
Channel bonding
  See Ethernet link aggregation
Chassis, 16–17
Compiler commands, 24
Compilers
  C and C++, 23
  Fortran, 23
Compute blade, 16
Compute node, 15
  dual-core processor, 9
Compute partition, 15
Compute Processor Allocator (CPA), 27, 37
Control Status Register (CSR), 41
Controller
  L0, 39
CPU
  Opteron processor, 9
Cray LibSci, 26
  FFT interface, 26
Cray MPICH2, 25
Cray SHMEM, 25
Cray XT series compute nodes, 1
Cray XT series systems, 1
Cray XT3 compute node, 1
Cray XT3 systems, 1
CRMS, 39
  actions, 43
  event handling, 44
  event logging, 44
  event probing, 43
  hardware, 39
  network, 40
  software, 41

D

Database nodes, 37
Debugging, 30
Development environment, 2, 22
Development tools, 2
DIMMs, 12
Disk storage, 3, 18
Dual-core processor, 9

E
ECC, 2, 9
Ethernet link aggregation, 3
Etnus TotalView, 30

F
Failover Manager, 41
Fibre Channel, 18
Fibre channel interface, 16
File system
  Lustre, 2, 34
  Object Storage Target, 34
  UFS, 2
Fortran commands, 23
Fortran interlanguage communications, 23

G
GCC compiler commands, 24
GCC compilers, 23
GigE interfaces, 16

H
Hardware components, 9
Hardware controllers, 41
HyperTransport Link, 13

I
I/O nodes, 21
I/O service nodes, 18
I/O system, 18
I/O test tool, 3
IEEE floating-point operations, 9
Interactive jobs, 27
Interconnect, 2, 14
Interlanguage communications, 23
IP
  native, 4
IPPO, 4

J
Jobs
  batch, 27–28
  interactive, 27

L
L0 controller, 13, 41
L1 controller, 41
Libraries, 25
  AMD Core Math Library (ACML), 25
  BLACS, 26
  BLAS, 25
  Cray FFT interface, 26
  Cray MPICH2, 25
  Cray SHMEM, 25
  FFT, 25
  FFTW, 26
  glibc, 26
  LAPACK, 25
  math, 25
  ScaLAPACK, 26
  scientific, 25
  SuperLU, 26
LibSci, 26
Linker, 26
Linux, 21
Logging in, 23
Logical machines, 3
Login load leveling, 23
Login nodes, 21
Login service nodes, 18
Lustre
  parallel file system, 34, 36
  shared-root file system, 33
  UFS-like file system, 34–35
Lustre file system, 22

M
Memory, 12
  Direct Memory Access (DMA), 13
  ECC, 2
Memory model, 2
Memory protection
  ECC, 9, 12

Message passing
  Portals interface, 13
Metadata server (MDS), 34
Microkernel, 22
Module, 16
Module (processor)
  See blade
Modules utility, 23
MPP, 1

N

Native IP, 4
Network
  CRMS, 3
  node-to-node communication, 2
  system interconnection network, 14
Network nodes, 21
Network router, 13
Network service nodes, 18
Node, 1, 14
  compute, 15
  service, 15

O

Object Storage Target (OST), 34
Operating system, 2, 21
  Catamount Virtual Node (CVN), 2
  UNICOS/lc, 2

P

Parallel file system, 36
Parallel programming models, 2
PBS Pro, 28
PE, 14
Performance API (PAPI), 31
Performance counters, 9
Performance measurement, 31
PGI compiler commands, 24
PGI compilers, 23
Portals interface, 13, 22
Process control thread (PCT), 27
Processing element (PE), 14
Processor, 9
Programming environment, 2

Q

Qk (quintessential kernel), 2

R

RAID, 3
RAS (reliability, availability, serviceability), 2
Rebooting compute nodes, 3
Reliability, Availability, Serviceability (RAS), 39
Resiliency communication agent (RCA), 41
Router, 13
Running applications, 27

S

ScaLAPACK, 26
SeaStar chip, 13
Secure shell, 23
Service blade, 16
Service Database (SDB), 21, 37
Service node, 15, 21
  boot, 18, 37
  database, 37
  dual-core processor, 9
  I/O, 18
  login, 18
  network, 18
Service node classes
  boot, 21
  I/O, 21
  login, 21
  network, 21
  Service Database SDB, 21
  syslog, 21
Service partition, 15
Shared-root file system, 33
Single-core processor, 9
Single-system view, 3
SMW, 18
SMW software, 42
sockets

AMD Socket 940, 9
Storage
RAID, 18
system RAID, 18
Striping, 36
SuperLU, 26
SUSE LINUX, 21
Syslog, 37
Syslog nodes, 21
System administration, 33
System interconnection network, 2, 14
System Management Workstation (SMW), 18, 33, 41
System RAID, 18

T

TotalView, 30

U

UFS, 35
UNICOS/lc operating system
Catamount microkernel, 22
SUSE LINUX, 21
User environment, 23

W

Warmboot,
See Rebooting compute nodes

X

x86 instruction set, 9
`xtshowcabs` utility, 27
`xtshowmesh` utility, 27

Y

`yod` utility, 27